УДК 004.9

THE APPROACH TO USER INTERFACE DEVELOPMENT BASED ON ONTOLOGIES

Oleksandra Bulgakova

Mykolaiv V.O.Suhomlynsky National University, Nikolska str., 24, Mykolaiv, 54030, Ukraine

sashabulgakova@list.ru

Представлені методи організації та основні вимоги до інтерфейсу користувача, які дозволяють сформулювати і описати структуру програмного забезпечення для моделювання та візуалізації складних систем і процесів, як потужного апарату для вивчення різних явищ і проблем. Запропоновано підхід до розробки інтерфейсу на основі онтологій.

Ключові слова: Інтерфейс, онтології, індуктивне моделювання, додатки.

This paper presents methods of organization and basic requirements for the user interface enable the software design phase to articulate and describe the software tools structure for modeling and visualization complex systems and processes as a powerful apparatus for studying various phenomena and problems. Propose approach to the interface development based on ontologies.

Keywords: Interface, ontologies, inductive modeling, application.

Представлены методы организации и основные требования к пользовательскому интерфейсу, которые позволяют сформулировать и описать структуру программного обеспечения для моделирования и визуализации сложных систем и процессов, как мощного аппарата для изучения различных явлений и проблем. Предложен подход к разработке интерфейса на основе онтологий.

Ключевые слова: Интерфейс, онтологии, индуктивное моделирование, приложения.

Introduction

In recent years, computer modeling and visualization tools have evolved from solving problems to a powerful mechanism for studying various phenomena and problems. They are successfully used in areas such as systems analysis, design automation, computational tools work organization and computer networks [1].

Modern tools for user interface development does not fully satisfy the specified requirements, namely the high-level declarative specification, software automatic generation, architectural components reuse.

During the past years, ontologies have been used in information sources for numerous purposes, such as annotating resources for better information retrieval, integrating data from different sources and systems, and automatically coupling intelligent agents [2-4].

To solve this problem, paper propose approach to the interface development based on ontologies. In this article, want to shed light at this area and take a closer look at the various possibilities of enhancing user interfaces with ontologies. Had reviewed various projects and identified a number of purposes for which ontologies can be used on the user interface layer. Each of those purposes poses particular requirements to the ontologies and their use in the application.

1. Interface organization methods

Interface - a combination of software and hardware providing the user interaction with a computer [2].

The interface can be arranged as automatic, interactive and interactive tools for user actions support:

Automatic that run the task to perform, connect it with specific data and perform some maintenance procedures;

The dialogue, which refers to regulated information exchange between man and computer which performed in real time and sent to a joint solution of the specific problem. Each dialogue consists of separate process I / O, which is physically provide a link between the user and computer. The information exchange is carried out by sending the message. In response, the user receives a prompt or help information messages to be answered, orders require actions, error messages and other information.

Interactive. According to [2], interactivity the ability of information and communication system actively and diversely respond to user actions. Therefore, an interactive interface the interface that allows on-line effectively intervene in the simulation process to correct some program parameters for improving the particular task results, and helps to reflect the intermediate and final calculation results in the form of both text and graphic information for subsequent visual analysis.

The main requirements to the user interface:

It is known that one of the major factors influencing the efficiency of the application software use, it is a convenience interface. This practice is allowed to form a common user interface requirements:

- Adaptability means supporting the dialogue style that would allow the user work with not previously familiar product, and provides representation in the usual form, as well as broad hints about the possible actions. In other words, the interface should be user friendly;

- Personal defines the usege of user interface with the "individual" projects;

- Adaptability - change of the external interface specifications during operation;

- Visibility;

- Availability - access to the program at any time.

2. User interface model based on ontologies

User interface model must contain all the information about the user interface, which may be subject to change in its life cycle [6]. Specific user interface model created on the basis ontologies that describe information about each interface model component universal ontology. In general, each of the universal ontology descriptions for constituents interface model O=<Name, OR>, where Name name set, Name ={<n, {dn}>}, n term name, dn description (attribute) of the term. OR - ontological relationships set. Information formation for a particular user interface is reduced to the allocation of the appropriate subset O′ universal ontology O and clarifying the values of its properties. Thus, it is possible to generate information for any number of user interfaces based on a universal ontology model. The interface can identify the following universal ontology O= {D, E, A, S, C}, where D domain ontology, E ontology expressive means interface, A – Ontology application, S – Ontology script dialogue, C ontology connection, C=C1∪C2, where C1 ontology between D and E, C2 ontology between D and A. Accordingly, O′={D′, E′, A′, S′, C′}.

Thus, the user interface model IM= < D′, E′, A′, S′, C′>.

Thus, the user interface model for any application can be viewed as a following models set: the system domain concepts, expressive means of interface, application, script the dialogue, as well as correspondence between models of domain concepts and means of expression, between the models and the application of concepts programs:

- user system concepts. In this system, the concepts expressed in the input and output data of the application program, and information about the intellectual support of the user actions.
- information system concepts. This class contains three types of concepts (in the future the number of such systems is expected to increase concepts): system graphical user interface concepts, the graphical static scenes paradigm and system concepts to form texts. Thus, each of the systems maintains design concepts of one type dialogue.
- the system concepts to determine the script dialogue. It defines the abstract terms to describe reactions to events (sets of actions to be performed when the event occurs, event sources, the view mode transitions between windows, methods of selecting copies of windows and others.).
- the system concepts in terms of which the connection between the application and the user interface. It defines the variables, the types of values that are common to the interface and application software, as well as the protocols by which communication takes place, server addresses, which carried out the connection, communication techniques.

Example tasks that will decide the interface [7]:

1. *The formalization input data* (retrieving data from external sources, their transformation and loading into storage);

2. *Data mining* (using inductive modeling algorithm: GIA GMDH)

Documents clustering will be made on the basis of generalized iterative algorithm inductive modeling (GIA).

Let us briefly consider the iterative structure of algorithm used for solving the general problem of search for a better model under such formulation:

$$f^* = \arg\min_{f \in \Phi} CR(y, f(X, \hat{\theta}_f)), \tag{1}$$

where $\hat{\theta}_f$ is an estimation of parameters for any partial model f∈Φ , CR is a model quality criterion for selection of optimal model.

The set Φ of models being compared can be formed by various generators of model structures of diverse complexities. All structure generators developed within the GMDH framework naturally divided into two main groups – sorting– out and iterative ones which differ by techniques of variants generation and organization of search of a given criterion minimum. For simulation will be used the generalized iterative algorithm, GIA GMDH, [8].

Formally, in the general case for layer *r* define the GIA GMDH as follows:

1) the input matrix is $X_{r+1} = (y_1^r, ..., y_F^r, x_1, \dots, x_m)$,

2) apply the operators:

$$y_l^{r+1} = f(y_i^r, y_j^r),\ l = 1,2,\dots,C_F^2,\quad i,j = \overline{1,F} \tag{2}$$

and

$$y_l^{r+1} = f(y_i^r, x_j),\ l = 1,2,\dots,Fm,\ i = \overline{1,F},\ j = \overline{1,m} \tag{3}$$

with a quadratic partial description

$$\begin{gathered} z = f(u,v) = a_0 + a_1 u + a_2 v; \\ z = f(u,v) = a_0 + a_1 u + a_2 v + a_3 uv\ ; \\ z = f(u,v) = a_0 + a_1 u + a_2 v + a_3 uv + a_4 u^2 + a_5 v^2\ . \end{gathered} \tag{4}$$

3) for each description is the optimal structure (an example for the linear partial description):

$$f(u,v) = a_0 d_1 + a_1 d_2 u + a_2 d_3 v\ , \tag{5}$$

where $d_k, k = 1,2,3$, $d_k = \{0, 1\}$ are structural elements of the binary vector $d = (d_1\, d_2\, d_3)$ taking values 1 or 0 (inclusion or not a relevant argument). Then the best model will describe: $f(u, v, d_{opt})$, where

$$d_{opt} = \arg \min_{l=\overline{1,q}} CR_l, \; q = 2^p - 1, \; f_{opt}(u,v) = f(u,v,d_{opt}) \qquad (6)$$

4) the algorithm stops when the condition $CR^r > CR^{r-1}$ is checked, where CR^r, CR^{r-1} are criterion values for the best models of (*r*–1) th and *r* th layers respectively. If the condition holds, then stop, otherwise jump to the next layer.

Define the GIA GMDH as many iterative and iterative combinatorial algorithms, described by vector of three elements DM (Dialogue Mode), IC (Iterative Combinatorial), MR (Multilayered Relaxative), ie any iterative algorithm is defined as a special case of a generalized: GIA (DM, IC, MR). This is possible with the help of specialized program complex of modeling based on iterative algorithms group method of data handling, which implemented the following features: automatic and interactive options for organization of user interface, management through the web interface, ensuring multiaccess. Constructed best model are presented by system for the graphic and semantic analysis, determined the effect of the arguments on the target factor, as well as analyzes and selects the most informative arguments [5].

3. *Information Definition to a particular ontology* (ontology instance).

After GIA finished will be obtained "ontology model". At this step, the text information will be analyzed with the help of the models obtained for each ontology (ontology instance) sorted. Each model will have its own threshold (minimum and maximum) value based on the error simulation. Thus, as a result of the phase is determined not only set the partition areas of knowledge, which will include text, but also the conformity degree of the relevant sections document, which gives reason to stop or continue the analysis.

4. *Creating a new ontology* (copies) on the basis of the analyzed information.

At this stage, we have the opportunity to create new instances of ontologies, which are not in the current knowledge base. After the formalization of the input data and analysis can remain documents that were not related to any category. Such documents will be stored in a special data warehouse and analyzed at regular intervals on the basis of which will constitute a glossary of terms. The dictionary will be stored semantic information, which will link elements of the dictionary, highlighting at the same time a new class of problem and domain.

3. Conclusion

Presented in the paper methods of organization and basic requirements for the user interface enable the software design phase to articulate and describe the software tools structure for modeling and visualization complex systems and processes as a powerful apparatus for studying various phenomena and problems. Paper propose a new approach to the interface development based on ontologies.

References

1. Bulgakova O.S. The basic approach to the interface software tools organization for modeling and visualization // Proceedings of the international scientific-practical conference « Information and communication technologies in science, education and manufacturing », October 29-30, Moldova, Tiraspol 2012. – C. 48-50.

2. User interface – http://ru.wikipedia.org/wiki/ User_interface

3. Ankolekar, A., Krotzsch, M., Tran, T., & Vrandecic, D. (2007). The Two Cultures: Mashing Up Web 2.0 and the Semantic Web. In C. L. Williamson, M. E. Zurko, P. F. Patel-Schneider, & P. J. Shenoy (Eds.), WWW (pp. 825{834). ACM.

4. Antoniou, G., Franconia, E., & van Harmelen, F. (2005). Introduction to Semantic Web Ontology

5. Conference on Human-Computer Interaction (p. 108-112). Lawrence Erlbaum Associates. Garca-Barriocanal, E., Sicilia, M. A., & Sanchez-Alonso, S. (2005). Usability evaluation of ontology editors. Knowledge Organization, 32 (1), 1-9.

6. Gribova V., Kleshchev A. From an ontology-oriented approach conception to user interface development International // Journal Information Theories & Applications. – 2003. – Vol. 10, № 1. – P. 87-94

7. Bulgakova O.S. Text information ontological analysis in the computer simulation systems // Inductive Modeling of Complex Systems. – K., vol. 6. – 2014. – P. 5-10.

8. Stepashko V.S., Bulgakova O.S. Generalized iterative algorithm of the group method of data handling // USiM. – 2013. – № 2. – P: 5 - 18.